Carleton University

COMP3004 Final Project

Neureset- Developing and Testing a Software-Based Prototype of a EEG Neurotherapy Device

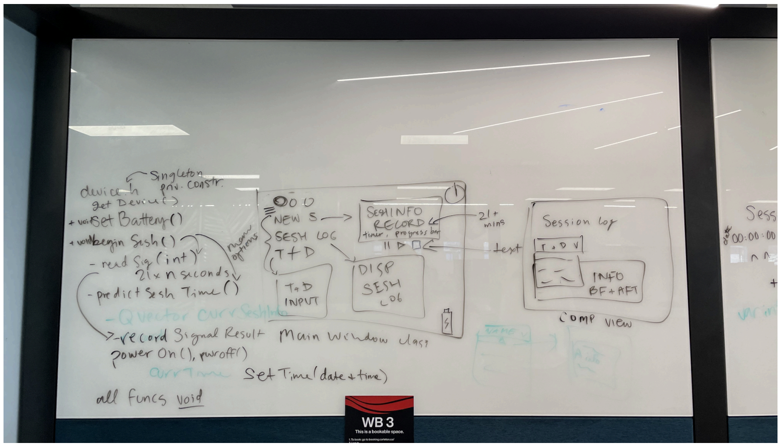
Team Members:
Ellie Ng
Ryushen Tan
William Yin
Stephen Mamosian
Mike Lin

Date: April 21st 2024

Design Decisions

Neurofeedback is a broad field that encompasses groundbreaking treatments for mental stress. For this group project, we created a simulation of a neurofeedback experience that could be the future--brainwave treatments at the press of a button in the comfort of a patient's home without the need for a computer or a medical professional. We implemented a GUI in QT C++ to showcase different use cases and functionalities of this product.

Firstly our user interface was carefully crafted with implementation and user experience kept in mind. By mixing a feeling of old and new we utilized UI UX design concepts such as wireframing to initially design a mock-up for how our implementation would look like using a whiteboard at school, then we decided a colour scheme as a group through our bi-daily group meetings. We made sure to add colour for buttons such as play and stop which helps the user identify functionality and implementation. We also utilized rounded corners and different button and background colours through the stylesheets functionality that QT C++ UI editor provides us to work with. We also provide convenient button and layout naming for easier backend implementation, as well as a more humanistic feel, for future developers who would like to keep the product up to date.



Initial Mockup of our Application

Additionally, another part of our design we provides a unique feature that allows us to view all the views at once in one display, so the session information, the computer view of the session, as well as the testing view so we can perform concurrent tests during our demonstration. This way developers and users can test out functionality within the UI. Users can also see status of the device through LED lights on the GUI that flashes or toggle depending on user device connection or device power.

For the EEG Signal generation functionality, we first tried to have a vector that stores the required amount of signals that were generated and then analyzes it. However, after we were more clarified on the requirements for signal processing and analysis, and after the *Testing Neureset* document it simplified what we needed to do for the complex signal processing part of this project. So, we decided to use Qt's signal and slots, and QTimer to constantly emit the EEG Signal values according to the sampling rate by creating the Signal Generator class. We saw that we only needed the frequency and amplitudes for the dominant frequency calculation, so we just made the Signal Generator class store all the frequencies and amplitudes, and we could set it to apply offset for the treatment. Thus, the only use of the actual EEG signal that is emitted is for it to be plotted. This lets us use Qt's observer design pattern as we just needed to have our EEGSite class and MainWindow class receive the EEG Signal, and easily handle them in their slots every time a signal was emitted as MainWindow needed to plot it. The EEGSite class handles all the signal processing as it calculates the dominant frequency by getting the required values from the Signal Generator class.

We have a Device class as the main controller of the whole simulator as it is responsible for running the session. Utilizing QTimer and the signal and slots, we also use it to set up the feature where users put on the headset, the QTimer sends out a timeout signal every second to update the round as we progress through the session. We also keep track of time and rounds. Using reference to the main window we can call front-end needed functions to make our backend program interactive with the front.

In conclusion, with our thoughtful design and implementation, this product is a good contender for scoring one hundo (hundred percent).